WHAT IS CLAIMED IS:

1. In a disk-based data storage system, a method for reference count regeneration, the method comprising:

sorting a virtual track table (VTT) to generate a reference list wherein the reference list contains each reference made by an entry in the VTT to an entry in a track number table (TNT);

counting the number of references in the reference list that refer to each unique entry in the TNT to generate a respective reference count list; and

merging the reference count list with the TNT to generate an updated TNT.

2. The method of claim 1 further comprising allocating a first memory region to hold the reference list, and allocating an additional memory region to hold further references that comprise the respective reference list when the sorting encounters an end of the first memory region.

3. The method of claim 1 wherein the sorting is a radix sorting that is performed using a bitwidth that is selected based on at least one disk cache attribute.

4. In a disk-based data storage system, a method for reference count regeneration, the method comprising:

radix sorting a virtual track table (VTT) to generate a reference list that comprises a plurality of sub-lists wherein each of the sub-lists contains references to a unique range of entries in a track number table (TNT);

counting the references in the reference list that refer to each unique entry in the TNT to generate a respective reference count list; and

merging the reference count list with the TNT to generate an updated TNT.

5. The method of claim 4 wherein the radix sorting comprises:

reading the VTT;

writing a first set of the sub-lists wherein each of the first set of sub-lists contains references to the TNT entries within a member of a first set of the unique ranges of entries in the TNT;

reading a selected one of the first set of sub-lists; and

writing a subsequent set of sub-lists wherein each member of the subsequent set of sub-lists contains references to entries within a member of a subsequent set of ranges of entries in the TNT, and the subsequent set of ranges comprises ranges that are within the range of the first set sub-list being read, and wherein the steps of reading the first set of sub-lists and writing the subsequent set of sub-lists are performed in parallel over a non-empty group of the first set of sub-lists.

6. The method of claim 4 wherein the radix sorting comprises at least one first pass radix sort that comprises:

reading a first unique portion of the VTT;

writing a first set of sub-lists wherein each of the sub-lists contains references to TNT entries within a member of a first set of the unique ranges of entries in the TNT;

reading a subsequent unique portion of the VTT; and

writing a subsequent set of sub-lists wherein each of the sub-lists contains references to TNT entries within a member of the first set of the unique ranges of entries in the TNT, and wherein the steps of reading the first unique portion of the VTT, reading the subsequent unique portion of the VTT, writing the first set of sub-lists, and writing the subsequent set of sub-lists are performed in parallel over elements of non-empty groups of the first and the subsequent unique portions of the VTT.

7. The method of claim 6 further comprising at least one subsequent pass radix sort that comprises:

reading the first set and the subsequent set of sub-lists; and

writing respective next subsequent sets of sub-lists wherein members of each of the next subsequent sets of sub-lists contain references to TNT entries within respective members of a subsequent set of ranges of entries in the TNT, and wherein the steps of reading a sub-list and writing the next subsequent set of sub-lists are performed in parallel over elements of the non-empty group of sub-lists.

8. The method of claim 7 further comprising:

counting references in each respective next subsequent set of sub-lists to generate a respective partial reference count list; and

merging the partial reference count lists to generate the reference count list.

9. The method of claim 8 wherein the step of counting the number of references in each respective next subsequent set of sub-lists is performed substantially simultaneously over elements of a non-empty group of the next subsequent set of sub-lists.

10. The method of claim 4 wherein the steps of counting and merging are performed in parallel over elements of a non-empty group of the sub-lists.

11. The method of claim 6 wherein a union of the members of the first set of unique ranges of entries in the TNT contains all of the entries in the TNT.

12. The method of claim 5 wherein the step of writing comprises:

allocating space in a memory region to hold a portion of a selected one of the sub-lists; and

comparing the size of the selected sub-list to the size of the space allocated to hold the selected sub-list, and allocating additional space to hold a subsequent portion of the selected sub-list when the size of the selected sub-list is equal to or greater than the size of the space allocated to hold the selected sub-list.

13. The method of claim 4 further comprising measuring a reference count list length, the measuring comprising:

selecting a subsequent set of ranges of entries in the TNT to be generated by a subsequent radix sort pass;

counting the number of references to TNT entries within each of the members of the subsequent set of ranges; and

allocating space to hold a sub-list to be generated by the subsequent radix sort pass, wherein the space is large enough to hold the number of references to the TNT entries within the member of the subsequent set of ranges that corresponds to the sub-list generated by the subsequent radix sort pass.

14. The method of claim 4 wherein the merging comprises:

reading a reference count value from the reference count list;

reading a TNT entry that corresponds to the reference count value;

comparing the reference count value contained within the TNT entry to the reference count value from the reference count list; and

writing an entry into a reference count mis-compare list, replacing the reference count value within the TNT entry with the reference count value from the reference count list, and writing the updated TNT entry back into the TNT when the reference count value contained within the TNT entry is not equal to the reference count value from the reference count list.

15. The method of claim 14 wherein the merging further comprises refraining from writing unchanged TNT entries back into the TNT when the reference count values within a block of contiguous TNT entries are equal to the corresponding reference count values from the reference count list.

16. The method of claim 15 further comprising refraining from writing the reference count mis-compare entries into the reference count mis-compare list when the number of entries in the reference count mis-compare list is equal to or greater than a first predetermined number of incorrect reference counts.

17. The method of claim 15 further comprising refraining from writing the reference count mis-compare information into the reference count

mis-compare list when the time interval for writing the reference count mis-compare values into the reference count mis-compare list exceeds a predetermined time or the size of the mis-compare values exceeds a predetermined size in the memory region.

18. In a disk-based data storage system, a system for reference count regeneration, the reference count regeneration system comprising:

a sorter configured to sort a virtual track table (VTT) to generate a reference list wherein the reference list contains each reference made by an entry in the VTT to an entry in a track number table (TNT);

at least one counter configured to count the number of references in the reference list that refer to each unique entry in the TNT to generate a respective reference count list; and

a merger configured to merge the reference count list with the TNT to generate an updated TNT.

19. The reference count regeneration system of claim 18 further comprising a first memory region allocated to hold the reference list, and an additional memory region allocated to hold further references that comprise the respective reference list when the sorter encounters an end of the first memory region.

20. The reference count regeneration system of claim 18 wherein the sorter is a radix sorter that is configured to sort using a bitwidth that is selected based on at least one disk cache attribute.